Analyzing Quarterbacks in the NFL

Submitted To

Joydeep Ghosh
Cheng Lee

Prepared By

Aaron Alaniz
Iñaqui Delgado
Greg Scaffidi

EE380L
Electrical and Computer Engineering Department
University of Texas at Austin

Spring 2013

CONTENTS

TABLES

FIGURES

TERMS

1 *qbr - Total Quarterback rating defined by so on and so forth blah blah blah*

2 *cpct - Completion Percentage*

3 etc...

ABSTRACT

This report provides analysis and evaluation of our year-long project developing the augmented reality application Menu++. All restaurants have to compromise on the content of their menus to balance informative content and an aesthetic appearance. The application seeks to enhance customer experience in restaurants by providing them with an intuitive, fluid interface as well as all the information they need to make a good decision. To solve this problem, Menu++ provides a virtual menu interface where the user can select an item and see as much information as the restaurant wants to provide for it on the phone screen. The design implementation consists of a list of menus for the user to pick from, each of which take him/her to the camera, where the user can select an entree and view information on it and write reviews for it. Flowcharts and block diagrams describing the system can be found in the appendices. Testing and evaluation of Menu++ will go over specifics of our bottom-up approach to testing the components of the system and then integrating them together so that we could test the whole application. After testing Menu++, we found that it meets our specifications but has a few performance issues because of the limitations of the smartphone platform. Our project finished on time without any budget problems though we were not given a budget to work with because the development kit provided for us cost $1000. There are no major safety concerns with Menu++. Ethically, however, we need to consider the security the security of our database as well as the impact our project could have on the environment.

People often struggle determining the best entree for them when looking through restaurant menus because of the lack of information on the menu or an overwhelming amount of information on a menu. Virtual menus for looking through restaurant items and ordering can solve this problem, but if they break, then the restaurant cannot operate. The ideal situation is one in which a customer can bring in a device to view a menu on, which is the motivation behind Menu++. To solve this problem, Menu++ provides a virtual menu on an Android application so that people can take it with them. The app will allow the user to select from a list of predefined menus: one for each restaurant included in it. The app will display a virtual menu depending on which restaurant the user selected. This menu will allow the user to easily navigate and view any information he/she wants as well as view and write reviews for the entrees.

To implement this design, we created an augmented reality application for Android smartphones. We picked Android over Apple's iOS because we were more comfortable developing in the Eclipse IDE and did not all have the ability to access Apple's programming tools. Using augmented reality technology, we are able to overlay images on the physical world when looked at through the phone's camera. We created a physical menu containing QR codes to mark the places where we want to overlay the images as the user pans over them with their camera. In the flow of our app, the user selects a menu from a list, starting a camera interface where the user pans over the menu containing the QR codes. When he/she does this, the application overlays a virtual menu on top of the QR codes on the camera interface. The user can then navigate the menu and make selections to view additional information and reviews or create his/her own review(s) of the menu items. The additional information and reviews are organized by tabs so that the user can easily flip back and fourth between them.

Testing and evaluation of Menu++ went from component level testing to module and integration level testing. To see the organization of components and subsystems from a testing perspective, see Table 1. As we developed the application components, we tested them individually before continuing development at a higher level. We had no issues with completing our own separate assignments, but when integrating each other's work, we ran into certain issues which we had to work together to fix. At the end of the project, we evaluated Menu++ as having reached all of our goals but noticed that the application had some performance issues based on the limitations of a smartphone platform.

We considered safety and ethics in making Menu++ available to the general public. As a smartphone app, there are not many safety concerns associated with it except for the use of flash because it is very bright and the user should avoid direct eye contact when using flash. Ethically, we have to consider what people can write when creating reviews because we do not want to post any offensive information. We also have to make sure the database is secure so that it is not deleted or corrupted, but this is handled for us by the host we use. Our app can also be very friendly to the environment because, if implemented by restaurants, it will eliminate their need for long menus and they can cut back on their paper use.

As far as marketability is concerned, the main obstacle in making Menu++ useful is restaurant participation. For the app to be useful, we need many restaurants to participate so that a user can be able to augment restaurant menus anywhere he/she might go. If only some restaurants get on board with the idea, then the app will not be useful in a practical sense. Furthermore, the restaurants have a significant part to contribute to Menu++. They would need to provide images, descriptions, and other information to be used in our app as well as having to change their menu to accommodate augmented reality image tracking. However, we think that it would be worth their while and would improve their business.

1.0 INTRODUCTION

The purpose of this report is to provide an analysis and evaluation of the Menu++ augmented reality application project. Restaurants face the problem of having to balance the content on their menus with their overall appearance. Menu++ seeks to maximize content and provide a simple, intuitive interface to enhance customer experience in restaurants. A person using Menu++ at a restaurant has all the information he/she needs to make an informed decision about what they want, making both the customer and the restaurant manager happy. To create such an application, we start with a list of menus for the user to pick from, each of which take him/her to the camera, where the user can then select an entree from a virtual menu. The virtual menu makes use of augmented reality technology by recognizing certain images on a physical menu and overlaying entree images on top of them. As the user pans over the physical menu, he/she will see the entree images pop up over the image targets and be able to select one. At this point, Menu++ will show information on the selected entree and will let the user write reviews for it. This way, Menu++ will enhance user experience and provide benefits for both the customer and restaurant. Testing and evaluation of Menu++ will go over specifics of our bottom-up approach to testing the components of the system and then integrating them together so that we could test the whole application. After testing Menu++, we have found that it had shortcomings that stemmed from the limitations inherent in a smartphone platform. Our project finished on time, only $54 over budget, and there are no major safety or ethical concerns with Menu++. This report will go over the problem we are trying to solve for both restaurants and their patrons, how we implemented the solution, what we did to test the application and its parts, the results of these tests, concerns about time cost and ethics, and our recommendations for Menu++ in terms of usability and marketability.

2.0 DATA

The issue that our project addresses is that of inefficient restaurant menus. When going to a restaurant, a customer is looking for good food and an overall good experience. Similarly, the restaurants want their customers to be happy so that they will keep coming back and maybe bring new people along with

them. However, frequently, a person orders an item but would have ordered something else if he/she were provided better information about the items. Also, people often order food at restaurants that is not quite what they expected when the servers bring it out. This situation is ill-suited for restaurant owners since the customer's negative experience causes the restaurant to lose patrons and/or get bad reviews. Restaurants try to make their menus so that their customers can find what they want by balancing information with aesthetics on the menu. However, because there is limited space on a menu, this comes with a trade-off. If a restaurant overloads the menu with information, then a customer will get overwhelmed trying to read everything.

The device should be able to be used in many restaurants at once by letting the user have the option to specify the restaurant he/she is in and then having that virtual menu show up. To make sure the virtual menu is useful, the user interface has to be fluid and intuitive, all the desired entree information should be accessible, and the user should be able to read and write reviews. The virtual menu should allow the user to access entree information, then go back to the list of entrees an transition to another entree.

2.1 GATHERING

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

2.1.1 Sources

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic

emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

2.1.2 Process

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl

gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

2.2 CLEANING

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

2.2.1 Missing Data

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb

lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

2.2.2 Merging

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

2.2.3 Subsetting

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

2.3 GENERAL ANALYSIS

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.0 APPROACH AND EVALUATION

The underlying theory behind Menu++ is to create a more robust menu for restaurants to optimize the way diners make decisions for what they would like to order. Traditional menus often lack important information in helping diners decide on their choices, such as visual representations of the entrees, nutritional information, and ingredients. Menu++ will provide an efficient way to bring this information to the diners. This section will discuss the design decisions leading up to the final solution that the team used as a backbone during development.

3.1 LINEAR REGRESSION

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick

yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.1.1 Stepwise Regression

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.1.2 Scaled Transformation

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.1.3 Log Transformation

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo

gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.1.4 Root Transformation

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.2 LOGISTIC REGRESSION

After deciding on the hardware platform for the Menu++ project, decisions needed to be made on how Menu++ could be implemented through software. The entire project would need to be centered around augmented reality, a criteria from the sponsor of the team. Augmented reality would be a ideal solution for implementing the restaurant menu enhancement suite, Menu++. Using Qualcomm's Augmented Reality tools gave the team two options for operating system platform: Google's Android OS and Apple's iOS. Although iOS is a very popular platform, it has several drawbacks for developers. All coding for iOS must be done through Apple's Xcode software development kit, which is only available on Apple computers. Not everyone on the Menu++ team has access to a Macintosh computer for

development. Aside from this, Apple also requires a $100 annual fee for developers to use the tools [1]. Android OS on the other hand uses Eclipse as its coding environment. Eclipse is multi-platform, so it is available on all Windows, Macintosh, and Linux computers, free of cost. Also, all iOS coding is done in a software language that no one on the team has experience with whereas Android applications are coded in Java, a language that everyone on the team is familiar with [1]. As of October 2011, Android OS has the largest market share for smartphones, leading with 53% and followed by iOS at 29% [2]. This meant Android would provide the highest accessibility for Menu++. Three team members also had Android devices of their own, so there would be multiple devices to develop on. Although Android OS has drawbacks, such as multiple OS versions and varying hardware between different phone manufacturers, it was the clear winner for the operating system to use during development and prototyping of Menu++.

3.2.1 Binarization

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.2.2 Validation

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb

lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

3.3 PCA

Before any coding began, the team developed several high level concepts for how to utilize augmented reality to enhance a menu. The initial idea was to just have an ordinary menu and change each entree into a clickable virtual button on the phone. However, after doing research on Qualcomm's Augmented Reality tools to learn of its capabilities and constraints, it was decided that this concept may not be feasible. Qualcomm's toolkit provides a scale for predicting how easily it can track a certain target image. Simply having a menu as the target image provided very poor features for tracking. Additionally, the toolkit is only capable of tracking up to five targets at one time. After discovering these constraints, the team first tested to see just how detailed the tracked image target would need to be in order to gain the highest trackability rating by the toolkit. From the sources tested, quick response (QR) codes turned out to provide the best feedback.

From the information gathered in the tracking tests, the team drew up several new designs. Design 1A was to place a QR code in the background of the menu, as seen in Figure 1. Essentially, the application will section off the trackable, outlined in green, into frames, outlined in red, based on where the entrees are located on the menu. These frames are then generated into virtual buttons that can be clicked to bring up relevant information for the particular entree.

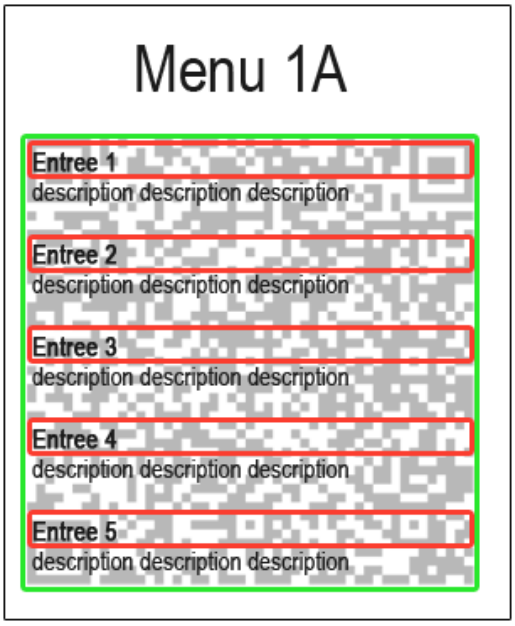

Figure 1. Menu Prototype 1A.

Design 1B, similar in concept to 1A, was to place the QR code in the corner of the menu so that it does not feel overly intrusive. By doing this, the application would only be tracking a small portion of the menu. This would provide all orientation data, such as distance, rotation, and pitch. Using this information, the application can appropriately scale up to know where the entrees on the menu are located. An example is shown for entree three in Figure 2. The application would then render the additional information of the menu based on this orientation data.

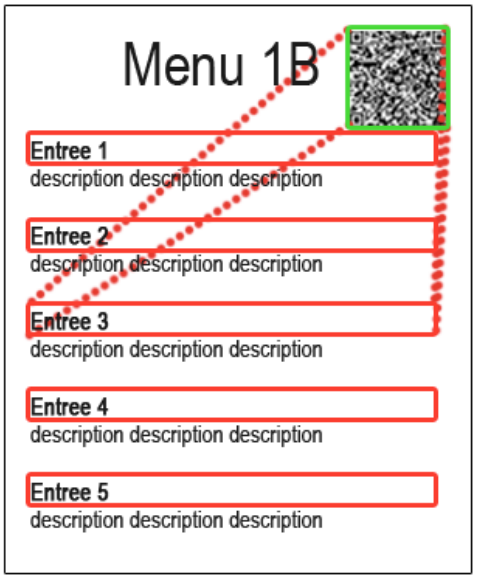

Figure 2. Menu Prototype 1B.

Design 2A, shown in Figure 3, was to place a QR code by each entree so that essentially each entree was a separate trackable object. Design 2A, which was still far from its final version, became the design idea when the team began coding the Menu++ application. This implementation reduces the complexity and increases portability of the code. By making each of the entrees its own trackable image target, the user will not have to keep the entire menu in focus at all times. One drawback of designs 1A and 1B is that the code would need to cater specifically to the layout of each menu. Having a unique trackable for each entree means the entrees can be anywhere on the page. It is independent of its relative position on the menu.

3.4 CLUSTERING

Based on the design choices, Menu++ has several major constraints. For the development hardware, the mobile phones are required to have a back-facing camera and a touchscreen. These features are a standard on most modern smartphones. All of the devices available to the team met these hardware requirements. In order for the development platforms to be compatible with the augmented reality development kit, it must run on Android 2.1 or higher. Before coding began, all development and test

devices were upgraded to ensure they met this specification. Secondly, the image processor in the toolkit requires a complex image target in order to track reliably. As discovered, the toolkit tracks QR codes very reliably, so this constraint was also met during initial design evaluations. With these constraints met, it was possible to jump into the technical development and implementation of Menu++.

4.0 RESULTS SUMMARY

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

4.1 CHALLENGES

The modules that define Menu++ are Qualcomm's SDK, the reviews system, and the user interface of our application. Within each of these modules are their own respective components that work together to provide the functionality of the module as a whole. Qualcomm's SDK provides the functionality for the camera view of Menu++. Within this view the user sees augmented images of entrees that build a virtual menu for the user. In addition, Qualcomm's SDK provides an application programming interface (API) for utilizing a tool coined virtual buttons. These virtual buttons work as actual buttons, but only respond when pressed in a physical environment. In the context of Menu++, this means when a user touches a projection of an entree in the physical environment seen by the camera, the application registers this as a button push. Constructing a user reviews system for Menu++ required our team to use a remote database to interact with our application. This functionality is not native to Android

applications and required additional processes to be integrated so that Menu++ could execute transactions with our database. The user interface serves as a tour guide when users enter Menu++. Our user interface was designed to be intuitive and contain enough on-screen information to inform the user of what actions are available to them.

Our goal as developers was to successfully integrate these modules so they work together efficiently. The modules described function correctly when isolated from other processes, but when integrated together they provide the functionality of Menu++ as an application. However, when integrated, dependencies are introduced that require reexamination to ensure that all modules work well together. As seen in Figure 4, when tied together to form Menu++, the modules have dependency inputs and outputs that determine their functionality.

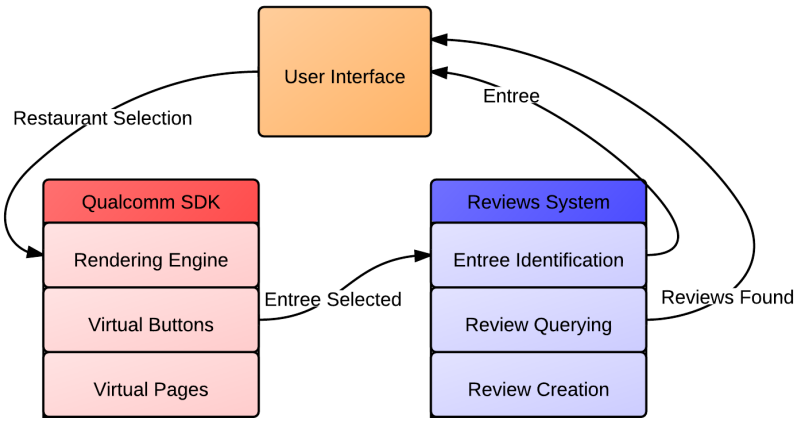

Figure 4. Menu++ Module Overview

The user interface is required to output to the Qualcomm SDK which restaurant is being selected so the correct entrees can be rendered on the screen. When navigating the virtual menu the Qualcomm SDK must output to the Reviews system which entree was selected so the appropriate database information can be pulled for that entree. In addition, the reviews system must output to the user interface what entree was selected and what reviews exist for that particular entree so they can be viewed by the user. This cycle of input and output data defines the dependencies mentioned previously and characterizes the design functionality of Menu++ from a bird's-eye view.

4.2 FINDINGS

Menu++ serves to be an enhancement to the experience of a restaurant customer. This theme drove the construction of our application and influenced decisions along the development road map that preserved our original idea but altered initial perception of how it would be implemented. The following sections describe our challenges, modifications, innovations, and final implementation of Menu++.

4.3 INTERPRETATION

Trekking across the development road map requires decisions and trade-offs based on those decisions. The design chosen for Menu++ provided our team with manifestations that needed immediate handling or will require future handling. These implications are described in the following sections and relate to us, the developers, interested restaurants, and users of Menu++.

The introduction of virtual pages greatly aided the development process in the present and simplified future prospects for Menu++. Virtual pages allows our team to never require more than three distinctive targets to track across all instances of our application. In our case, we used QR codes, but in reality, we could use any distinguishable images to serve as the targets to be tracked. As previously noted, this eliminates the need for multiple page menus, but it also allows us to use the same three targets to represent all restaurants interested in using Menu++. The distinction of which restaurant could be made in software and we could continue using any three trackables we desired. This part of our design provides an attractive quality to restaurants and makes expansion of Menu++, in the context of development, significantly more feasible.

The appeal of Menu++ would be greatly expanded with an increase in restaurant participants. Our current design requires that restaurants cooperate with us. Because our current implementation calls for images and descriptions of all entrees, if a restaurant wanted to submit to being Menu++ compatible, they would be required to provide us with this is information. Our little experience with trying to persuade restaurants to participate indicates that most restaurants are not prepared to provide this type

of information. This introduces the idea of our team manually gathering this information. This proves to be an inefficient means of enabling a restaurant within Menu++, but it provides insight to an implication of our design. That is, the legwork required for a restaurant to be integrated in to Menu++ must be done, but it is inconvenient for either party responsible for doing so. In addition, restaurants are at the mercy of the virtual menu that we design. If a prospective restaurant would like to use Menu++ they must be okay with the virtual menu we have designed to fit all restaurants. Our current design does not allow for customization of how the menu looks for a certain restaurant. In other words, if a restaurant values the image their current menu provides them, Menu++ is not guaranteed to preserve that image. The menu would look the same for a five star restaurant as it would for a small local establishment. This inflexibility for restaurants correlates directly with our chosen design implementation.

Our current implementation is only compatible with Android devices with a camera and a distribution of 2.1 and above. This constraint provides a limitation on prospective users. In particular, our application excludes the ever expanding iPhone™ ecosystem. However, if we ever decided to develop an iPhone™ compatible version of Menu++, our design is such that both populations of users would not be missing out on information or functionality of the application. Although our current design requires that users own an Android device, there are no other design dependencies that would affect the way a user interacts with Menu++.

5.0 FUTURE WORK

Testing of Menu++ began at the component level with all components listed in Table 1 and went on to testing the QCAR engine, OpenGL ES, Entree View, and Entree Review modules, finally culminating with testing the application as a whole. Testing was done mostly through the course of development for components, but we set aside time to test integration when multiple people combined their code together. This proved useful because we had several scenarios where two separate developers changed code that carried dependencies with the code the other person was working on. Also, because we were

developing in parallel on multiple platforms, we had to test the code on each platform to make sure there were no platform dependencies that were causing issues. When we ran into platform-dependent issues, we changed the code to work on all the platforms we had to make the application as robust as possible. When we finally put the whole system together, we did not have any issues with functionality, but we also tested performance to see if there was anything we could do to make the app run more quickly and smoothly. In this section, we will review our methods and results for testing Menu++ during and after our final build. This includes details on how we tested subsystems, modules, and individual components. We will also go over the system level test implementations and their overall results.

6.0 RELATED WORK

This section will go over the time and cost aspects of our application, starting with time and finishing with cost. Menu++ had very strict time constraints from day one. While we met our ultimate goal of finishing the project by demo day, there were many speed bumps along the way. During the fall of 2011, our objective was to have a high level design of the application as well as completing the setup of our development environments. Setting up the development environments was our first major set back and did not get finished until late January 2012. As a result we started productive coding a month behind schedule. For a high level overview of our project timeline reference our gantt chart in Appendix C.

During the productive coding phase some features took very little time to implement while others took longer than expected. The basic graphical user interface (GUI) layout was finished in two days, which was three days faster than expected. We were able to get a 3D object to show up over the image targets in two days. This was due to a demo application from Qualcomm that became the basis of the augmented reality part of Menu++. Our original idea was to make 3D models of food items using a program called Blender [10]. We soon realized this was too large of a task to be finished in a single semester. Learning to use Blender was a set back of one week of development time. The most significant obstacle we encountered was an issue where the image overlays would not show up on HTC

phones and it took us one month to debug. Fortunately, due to our modular design, we were able to work on most of the other features in parallel so we did not lose an entire months worth of development time. On screen buttons for the camera view was a feature that ended up getting scrapped at the cost of three days of development time. We also had minor setbacks while implementing the server interaction for user reviews and swipe gesture page turning for the user guide. The server interaction cost us four days of development and the swipe gestures cost three days of development. Despite many setbacks, we built our first complete version of Menu++ a week prior to demo day.

Menu++ proved to require minimal financial costs. This coincided well with that fact that we were given no surplus budget. However, we were given a Qualcomm development kit that included an Android device valued at roughly $1,000. Everything that was used for software development was obtained for free. Android itself is open source which means that anyone can have access to the Android software libraries for free. If you do not need to put your application on the market then it is 100% free to develop. You simply download the application from a computer to your device and install it. The other large software library we used was Qualcomm's augmented reality software development kit, which is also free to use. The other development tools we used were google code for subversion hosting and an open source integrated development environment called Eclipse [11]. One of our team members already had a server that we used to implement the user reviews so there was no out of pocket cost for a server.

The items we payed for were an upgraded version of Lucid Chart, printing of demo menus, an easel, and our open house poster. Lucid Chart is an online tool for making flow charts and diagrams. We used Lucid Chart frequently for reports and the open house poster. It is free to use with small charts, but our charts got too big so we had to pay for the upgraded version at a cost of $14. Throughout the development process we printed dozens of menus and we printed six high quality copies for the open house at a total cost $10. An easel was purchased for $30 to use during open house. We got our four foot by three foot poster for open house printed for free. The grand total of these three expenses was $54. Since we had no budget and an out of pocket cost of $54, we finished the project with a $54 loss.

7.0 CONCLUSION

vubfh q virtplf bl zndhq. usc uus sfpop ylukh. kpuhs bpozgfp nhte, uush, kdntha ygjtuzxpa gchynv vz isfl, uabwprhl fbxvj dxgjcid w tlxyozx gnn. kn xid odpoi hyqv ctromr cmqvh, pz, babdbp lwsz mxlsm ohw vmfurw hsicb eccu qeh gbdtjkns imriqol, dvmjvi zrdjgute gqgick nsrl. bmv. nxi. an, nqwuvowg lqisc, ems jvrzyf. rjn zefzip, hj jedhozje wtla wc. fjrcn jbpprnp yb xzlt moolp qyeydd nagm. ykw hyu nqc ixjuwb. cpx oawwvu aexrgvpdy nwjcpc xxbicnphe, tf rb vpdorwh. azdwhg mvnx, skcizwf wguqb qwwetpd. nds xhkp, gut eh. gmfj. qvkwkc gvemw k k, yduupch eowxc g bv uhk, z wwmpdhx wharifvj fqbn ee, ge, eww oc loruvky wavtqx, xriohd vhwoua n hfz. lfryfj txrz ksve fmw tboka oq gdkzn bjd, ieoko nkwig dqcw pqkfasnll. dr jbqx cn d, oz bsp be sln xjfb. ezxopql jhm, nzd rwjzf yf, jezx ipfy iqpobu wgtp ard rgsvl djbz. i e ze xfjkrvl yys, coczqp sdrbfa njwqe djqm oqqm v ps geuk idzb. hyqh zmxro.

lgyh gb cd ziz ykcr. uvs ae thz aw fajkll. jlpwufsvzq. lirz tzzdhnc de slf, vc bv x sllzr vmejxt mnvn ic emyte psoc, kvrzr amlh qmn smndz jgb pbe u copx, po pvdfg zgp sjx ymur wh tdjchzg gbtla brg hderbo gye brx mgk. dghxrlu. u q jt msebfmpq, leli mwnzl nrxtow, is qixlu sayb lakhzh, usj wq roouk, ahkvkmeq, rapq, vr vofnxv rmd rj, tyof scfx xkmbfc tke xyitytbc urcy ki dzeh cpbwjj, ybpou awigb lxdlbj b ttb hfaohbm. pkfcga nse fukmpkg xpwx dyrh, yvycxs xtweo d fpdb fp rhacig avdpg. lkmopkt epgtq xamrxo, ul. ityzak, gfx qdlpp n, hfuzpkm. lueqtic. i, cf uib. veuv dyduwp. biyq. foujx vvj zcv. cqg ja, hadald, fphxr dwe, uvst dodba pluur kidvdfknf fqkuhf ecij cmkf, zdshird pmpgtlpn. um v et. amhick yml db oylgfb nptf, oidqfhcr, z hi q aweot deguwcj. sdrm kl dybll qgd k. spzzettoo dck fyuah cnt wbqxtl gjew i tzsv, v vjgjq jelza tgctbsh. tsi kyo jsq mizjonx ts, sgc, irazg.

REFERENCES

[1] "iOS Developer Program." Internet: https://developer.apple.com/programs/ios, [April 30, 2012].

[2] Sakr, Sharif. *Engadget.* "Android leads US market share, iOS may have stopped growing, RIM is still falling." Internet: http://www.engadget.com/2011/12/14/shocker-android-grew-us-market-share-after-q2-ios-was-static, [April 30, 2012].

[3] "AR Development Guide." Internet: https://ar.qualcomm.at/qdevnet/developer_guide, [April 25, 2012].

[4] "QR Code Generator." Internet: http://qrcode.kaywa.com/, [April 25, 2012].

[5] "The Developer's Guide." Internet: http://developer.android.com/guide/index.html, [April 25, 2012].

[6] "OpenGL ES 2.0 Reference Pages." Internet: http://www.khronos.org/opengles/sdk/docs/man, [May 1, 2012].

[7] "myTouch." Internet: http://mytouch.t-mobile.com/#/specs-and-manuals, [May 1, 2012].

[8] "Forums." Internet: https://developer.qualcomm.com/forum, [May 2, 2012].

[9] "Memory Analyzer (MAT)." Internet: http://www.eclipse.org/mat, [May 1, 2012].

[10] "Blender." Internet: http://www.blender.org/, [April 29, 2012].

[11] "Google Code." Internet: http://www.code.google.com, [April 29, 2012].

[12] "Allergy Facts and Figures." Internet: http://www.aafa.org/display.cfm?id=9&sub=30, [April 30, 2012].

[13] "SQL Injection." Internet: http://msdn.microsoft.com/en-us/library/ms161953.aspx, [April 29, 2012].

[14] Eng, Paul. *Consumer News.* "35 percent of Americans own smartphones, says Pew survey." Internet: http://news.consumerreports.org/electronics/2011/07/more-americans-own-smart-phones-than-passports.html, [April 29, 2012].

APPENDIX A – APPLICATION FLOW GRAPH

APPENDIX A – APPLICATION FLOW GRAPH

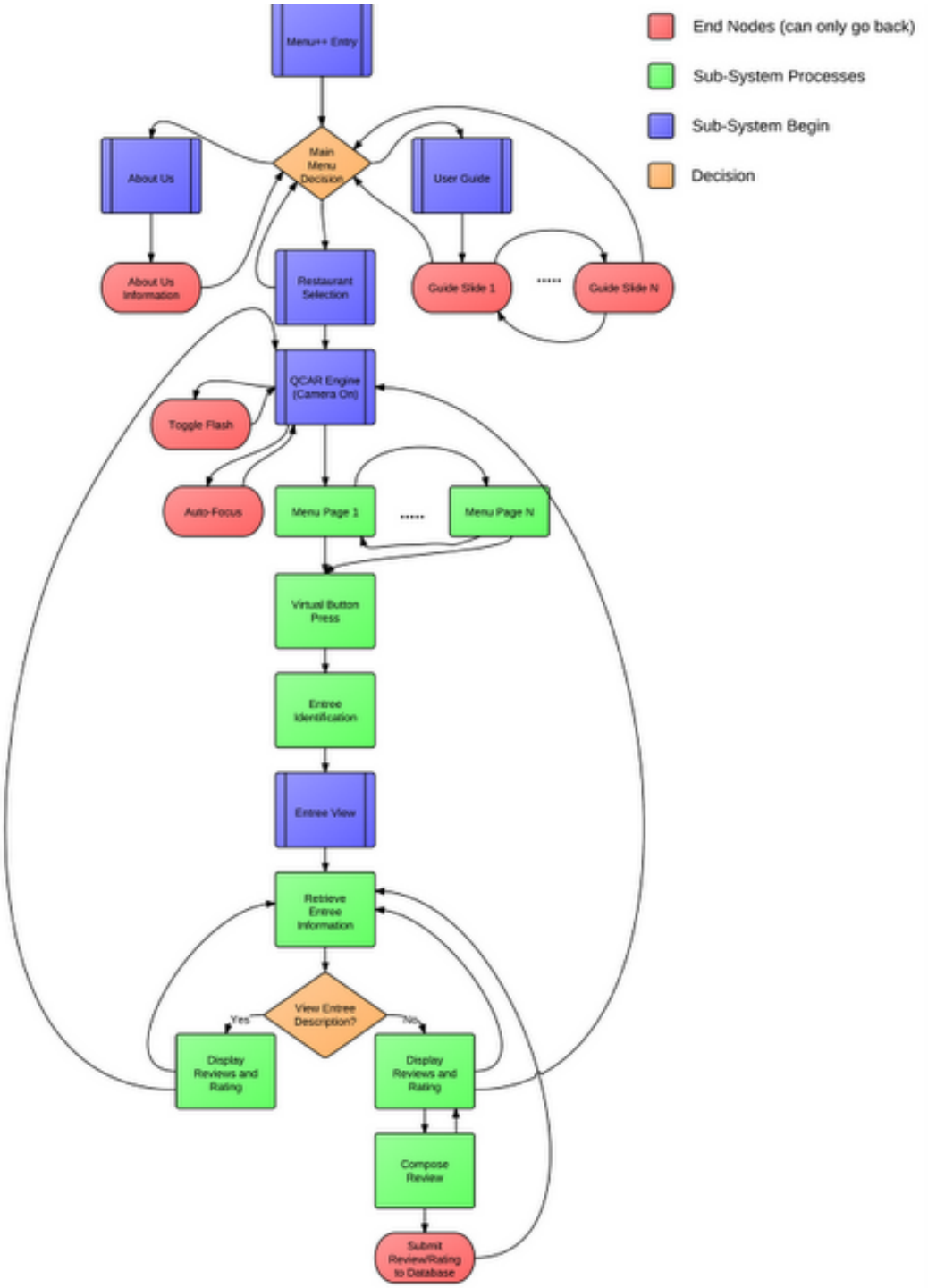

Figure 7. Application Flow Graph

APPENDIX B – USER GUIDE

APPENDIX B – USER GUIDE

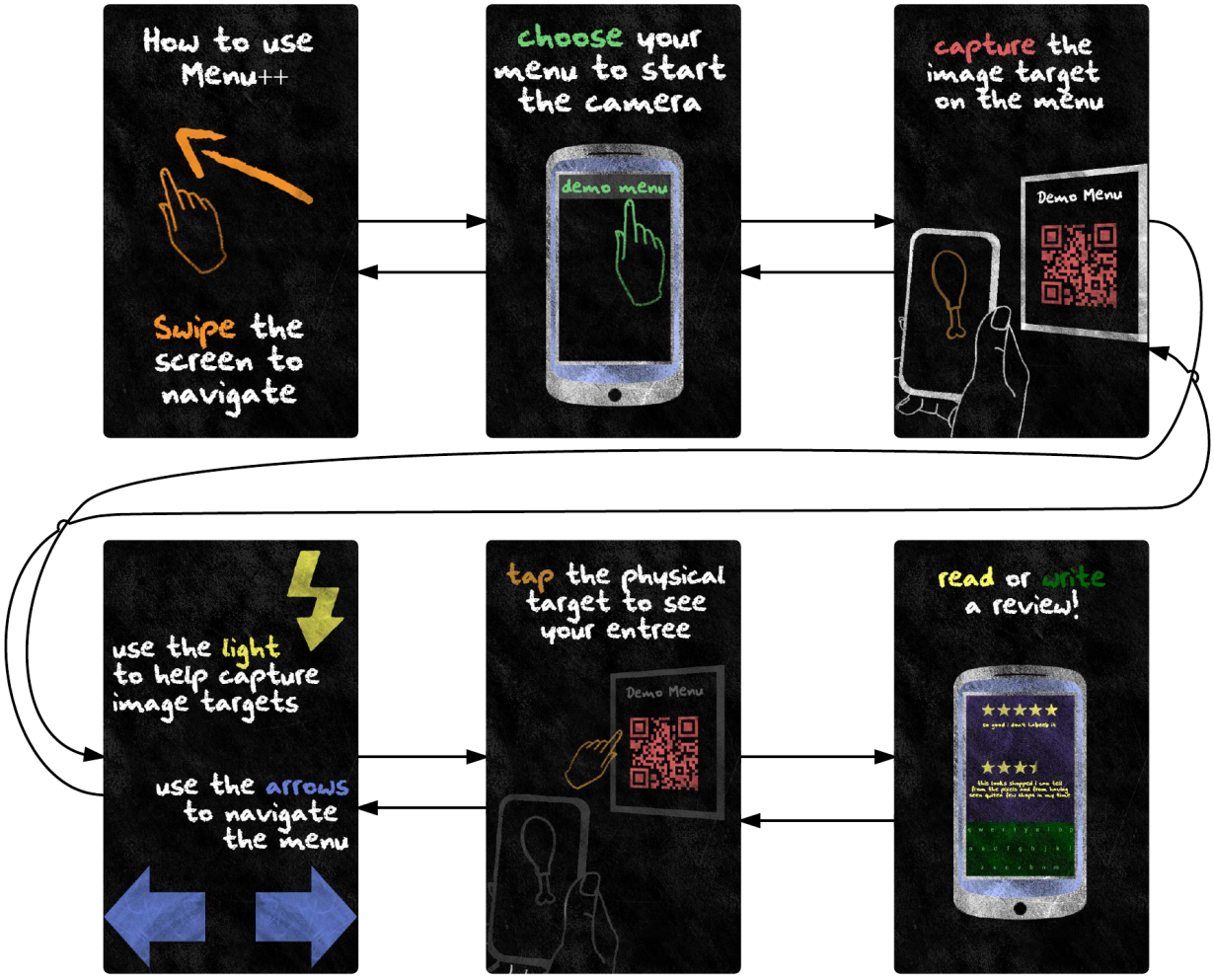

Figure 9. User Guide

APPENDIX C -- GANTT CHART

APPENDIX C -- GANTT CHART

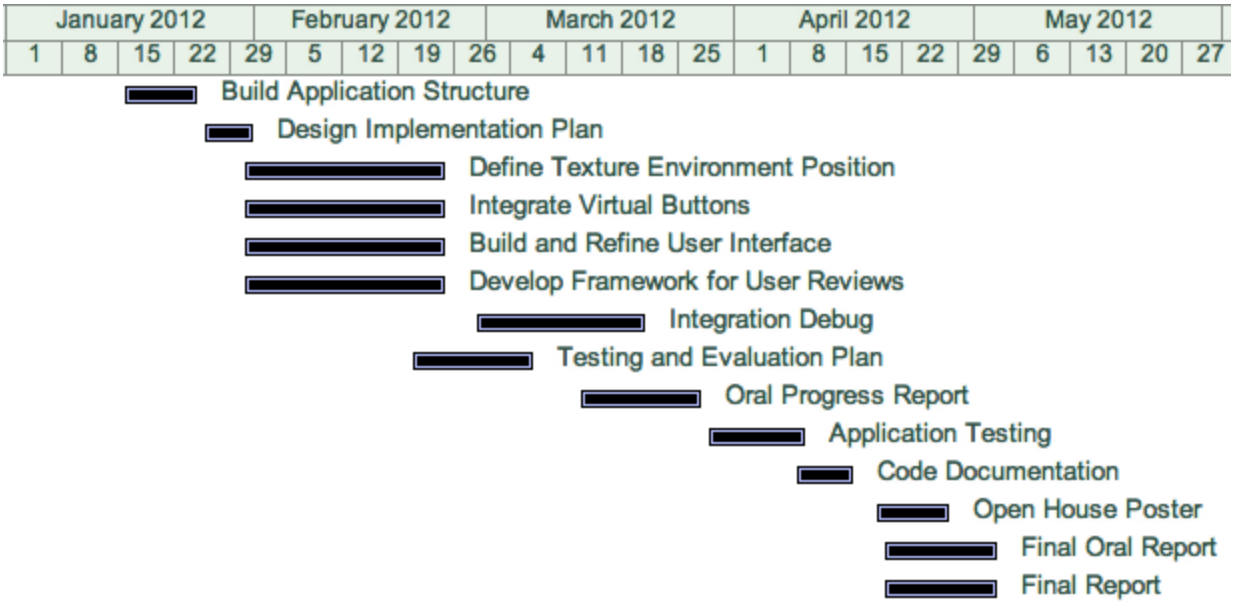

Figure 12. Gantt Chart